基于 GPU 加速的粒子滤波多说话人跟踪算法及其应用 *

曹　洁[a, b]，黄开杰[b]，王进花[b]

(兰州理工大学 a. 计算机与通信学院; b. 电气工程与信息工程学院，兰州 730050)

摘　要：为了解决粒子滤波多说话人跟踪过程中粒子易发散导致多目标跟踪精度低的问题，提出了并行粒子滤波和基于 GPU 的 K-均值聚类的多声源定位方法。该方法首先分析了粒子滤波在实现多目标跟踪时，进行数据关联的过程产生较大的计算量，并且出现多个目标时，粒子会逐渐发散。针对计算量大和粒子发散的问题，提出了一种并行粒子滤波和 K-均值聚类的方法。实验表明，随着粒子数和目标数的增加，计算量以指数增加，并且粒子发散严重，采用基于 GPU 的 K-均值聚类方法的粒子滤波多说话人跟踪方法，相比传统粒子滤波跟踪方法具有更收敛的粒子集并且跟踪精度较高。

关键词：GPU；粒子滤波；K-均值；多目标跟踪

中图分类号：TP301.6

Particle filter multi-speakers tracking algorithm based on GPU and its application

Cao Jie[a, b], Huang Kaijie[b], Wang Jinhua[b]

(*a. College of Computer & Communication*, *b. College of Electrical & Information Engineering*, *Lanzhou University of Technology*, *Lanzhou* 730050, *China*)

Abstract: In order to solve the problem of low accuracy of multi-target tracking, particles in particle filter are easy to disperse in the process of multi-speaker tracking. This paper presented a parallel particle filter algorithm and GPU-based K-means clustering multi-source localization method. The method first analyzed the particle filter to achieve multi-target tracking, data association process have a large amount of computation, and the particles gradually diverge with the emergence of multiple targets. In order to solve the problem of large amount of computation and particle divergence, this paper proposed a method of parallel particle filter and k-means clustering. Experiments show that, as the number of particles and the number of targets increases, the amount of computation increases exponentially and the particles scatter seriously. In this paper, by using the GPU-based k-means clustering method, the particle filter multi-speaker tracking method has more convergent particle sets and higher tracking accuracy than the traditional particle filter tracking method.

Key Words: GPU; particle filter; k-means; multi-target tracking

0　引言

说话人运动过程[1,2]具有较强的随机性，通过采用合适的模型来描述说话人的运动过程，有研究人员提出用动态空间模型的方法描述说话人的运动，并利用滤波理论[3,4]解决说话人跟踪的问题，动态空间模型提供了说话人运动规律的先验信息，利用先验信息有利于提高说话人定位的精度。在说话人的定位与跟踪中，卡尔曼滤波方法[5,6]解决了高斯线性模型下的系统状态估计问题；扩展卡尔曼滤波方法[7,8]时延估计作为观测量进行非线性滤波，解决了高斯非线性模型下的系统状态估计问题；无际卡尔曼滤波方法[9]解决了高斯非线性模型的状态估计问题。粒子滤波方法[10]非高斯非线性的系统模型，这更加接近真是场景，并将时延估计作为观测量，滤波后直接得到声源的位置，该方法解决了非线性非高斯模型下的系统状态估计问题。实际场景通常是非线性非高斯的，因此粒子滤波在说话人的定位与跟踪应用最广泛。后续研究中，许多研究者针对粒子滤波在说话人跟踪的缺陷提出改进，改进方向有三个，一是定位特征的提取[11]；二是粒子滤波算法的改进[12]，针对算法的不足提出改进并应用到说话人跟踪中；三是采用更合适的动态空间模型来描述多说话人运动过程。另外传统粒子滤波在多目标跟踪时，通过使用多个粒子滤波器，每个滤波器跟踪一个目标，利用数据关联方法将不同的量测值与目标进行相对应，从而获得每个

基金项目：国家自然科学基金资助项目（61633031，61763028）；甘肃省自然科学基金资助项目（1506RJZA105）

作者简介：曹洁（1966-），女，安徽灵璧人，教授，博导，主要研究方向为信息融合理论与应用、智能交通、信号检测与估计；黄开杰（1989-），男，湖北大冶人，博士研究生，主要研究方向为信息融合理论与技术（425449030@qq.com）；王进花（1978-）女，甘肃天水人，副教授，硕士，主要研究方向为检测技术与自动化装置.

目标的特征，每个目标的跟踪是一个独立的过程。关联过程会产生较大的计算量，并且粒子滤波在多目标跟踪中易产生粒子发散和跟踪精度低的问题。

针对计算量大的问题，本文引入了并行的思想，将粒子滤波并行实现；并且针对粒子滤波进行多目标跟踪过程中，重采样后的粒子易发散的问题，本文采用基于 GPU 的 K-均值聚类方法[13-16]，将 K-均值聚类算法中计算密集的部分交给 GPU 端执行计算。

1　说话人运动模型

多说话人运动模型用于表示移动的说话人在智能会议室环境里位置信息的变化，其中广泛采用的模型为 Langevin 模型。该模型综合了各类随机运动的特点，在声源的定位与跟踪问题中，该模型能够较好地描述说话人的运动过程，因此被广泛采用。

声源在 X 轴方向的运动方程如下

$$\alpha_t=[x_t,y_t,\dot{x}_t,\dot{y}_t] \tag{1}$$

$$\begin{aligned}\dot{x}_t&=a_x\dot{x}_{t-1}+b_xF_x\\x_t&=x_{t-1}+\Delta T\dot{x}_t\\a_x&=e^{-\beta_x\Delta T}\\b_x&=v_x\sqrt{1-a_x^2}\end{aligned} \tag{2}$$

其中: x_t 和 $\dot{x}_t$ 分别表示 X 轴方向的位置和速度， y_t 和 $\dot{y}_t$ 分别表示 Y 轴方向的位置和速度。 F_x 是正态分布的随机变量，$\Delta T=L/f_s$ 为帧间的时间间隔，L 是帧长，f_s 是采用频率，β_x 为常数，v_x 为稳态均方根速度，本文中 $\beta_x=10^{-1}s$，$v_x=10^{-1}ms$。

2　多目标模型

根据目标运动状态和数量的变化，目标的运动模型有四种：目标运动，目标消失，目标产生和目标衍生，如图 1 所示。

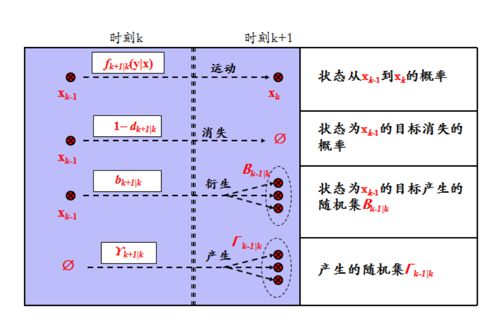


图 1　多目标的运动模型

2.1　基于粒子滤波和 K-均值聚类的多说话人跟踪方法

传统粒子滤波在进行多目标跟踪时，通过使用多个粒子滤波器，每个滤波器跟踪一个目标，利用数据关联方法将不同的量测值与目标进行相对应，从而获得每个目标的特征，每个目标的跟踪是一个独立的过程。关联过程会产生较大的计算量，并且传统粒子滤波在对多个目标进行跟踪时，随着目标数量的增多，粒子易出现发散的现象。针对粒子发散和计算量大的问题，本文引入并行策略，将粒子滤波与 K-均值聚类算法并行实现，从而减少计算复杂度，从而达到提高粒子滤波跟踪算法跟踪精度的目标。

2.2　基于 GPU 的 K-均值聚类算法

K-均值聚类算法核心思想是：首先从所给的 n 个对象中选取 k 个对象作为初始聚类中心点，对剩余的其他对象计算它们与所选的 k 个中心点的相似度，然后再重新计算所获聚类的聚类中心。算法基本流程如下：

a）从粒子集 X 中随机选择 k 个对象作为初始聚类中心 $c_1,c_2,\cdots,c_k$。

b）将对象 $x_i(i=1,2,3,\cdots,n)$ 按照距离分配给最近的一个聚类中心。 x_i 和聚类中心的距离定义为

$$\| x_i-c_j \|=\sqrt{\sum_{l=1}^{m}(x_{i,l}-c_{j,l})^2} \tag{3}$$

其中: m 为数据的个数。

c）重新计算每个簇中新的聚类中心 c_j

$$c_j=\frac{1}{N_j}\sum_{x\in S_j}x_i\quad ,\ j=1,2,\cdots,K \tag{4}$$

其中 N_j 为第 j 个聚类簇 S_j 中对象的个数。

d）重复 b）c）直到 k 个聚类中心不再变化，即准则函数收敛。

基于 GPU 的 K-均值主要思想是数据并行，将传统 K-均值聚类算法中计算密集的部分放在设备端进行计算，从而实现提高性能的目的。数据对象分配和 K-均值中心点的计算执行多次，并且不同的数据独立。将数据对象分配和 K-均值中心点计算可以分离成两个内核函数，该函数由大量线程组成，在主机端调用，在设备端执行。在任务分配时，主机负责将 k 对象放置到被聚类的对象所表示的空间，并重新安排所有的数据对象与控制迭代过程，而设备端进行数据并行密集计算。在数据对象赋值之后，得到每个数据点的簇标签，用于计算一个聚类中心。本文将聚类标签从设备端下载到主机端，主机端重排所有的数据并计算每个群集所包含的数据对象的数目。

算法 1　基于 GPU 的 K-均值聚类算法

if threadId=0 then
$c_{l_i}\leftarrow c_{l_i}+x_i$
end if
Synchronize threads
repeat
for all $c_{l_i}\leftarrow c_{l_i}+x_i$ do
　$c_{l_i}\leftarrow c_{l_i}+x_i$
end for
synchronize threads
if threadId=0 then
for all $c_{l_i}\leftarrow c_{l_i}+x_i$ do
　$c_{l_i}\leftarrow c_{l_i}+x_i$
　$m_{l_i}\leftarrow m_{l_i}+1$

```
for all  c_j ← (1/m_j) c_i  do
     c_j ← (1/m_j) c_i
end for
if convergence then
     signal threads to terminate
end if
until convergence
```

在任务分配时，主机负责将 K 对象放置到被聚类的对象所表示的空间中，并重新安排所有的数据对象与控制迭代过程，而设备端进行数据并行密集计算。在数据对象赋值之后，得到每个数据点的簇标签，用于计算一个聚类中心。最简单的思路是读取所有的数据对象和确定数据点的聚类中心。然而，大量的条件语句不适用于 GPU 处理器模型。因此本文增加一个过程，这个过程包括将聚类标签从设备端下载到主机端，主机端重排所有的数据并计算每个群集所包含的数据对象的数目。在数据存储时，数据对象和中心点以动态数组的方式在设备端储存。本文将 K-均值聚类算法的所有参数存储在在全局内存中，因为常数存储器和纹理存储器均为只读的存储器，且存储内存只有 64kb，内存太小以致于不够容纳 K 均值聚类的所有参数。

由于 GPU 是一种共享内存的处理器架构，处理器被称为线程，并且是主-从模型。每个线程都被限定在 0 至 t-1 之间，t 是线程总数。通常情况下，线程 0 被认为是主线程，所有其他的线程被认为是从线程（任何一个线程都可以被认为是主线程，其他为从线程）。线程共享数据点集所在的一部分内存，同样还包括当前的中心点集 n，还有计算所有内部中心点直接的距离 n 矩阵 A 和对该矩阵 A 进行排序，获得下标排列矩阵 B。每一个线程还附加了三种内存，供内部数据使用。假设对封闭的架构访问是可行的。映射到 GPU 的编程描述如下：首先主线程和串行算法一样初始化所有的中心点，然后 n 被分割成子集 n。这仅仅是标记阶段，每个线程执行赋予 n 单独的处理器的坐标和计算范围。所有线程执行标记阶段都是针对于它们各自的 n 划分。每个数据点 n 的标记都被存储在一个 n 维向量的组件 l_i 中。这样便消除了当更新聚类和简化记录过程中，同时写的问题。

在标记阶段之后，线程间需要进行同步操作，以确保所有中心点更新的阶段数据已经是可以使用的。之后，中心点更新阶段可以通过规约操作来执行。

然而，为了简化，假设主线程执行这个阶段是顺序执行的。主线程不是迭代全部中心点，而是迭代部分计算的新中心点的全部标记。一个 m_j 维向量 m_j 在一次迭代中被更新，向量的每一个元素 m_j 保存着数据点被分配到聚类 C_j 的数值。执行下一个循环遍历所有中心点，用 $1/m_j$ 处理 C_j 来获得最终的中心点。收敛判断同样是由主线程决定，主线程通过检查最好一个标记阶段在聚类算法中是否有改变来确定。只要达到收敛的条件，从线程由主线程来停滞其执行。

基于 GPU 的并行 K-均值聚类算法流程如图 2 所示。

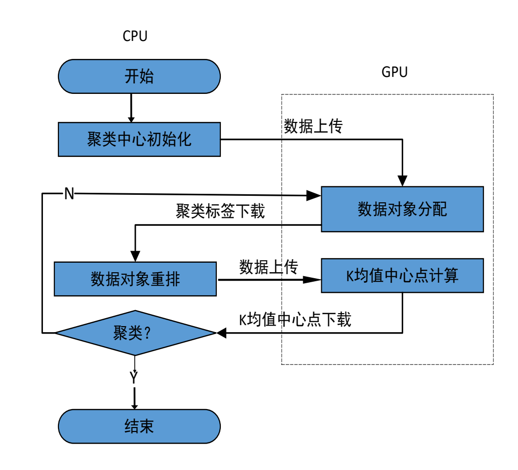


图 2　基于 GPU 的 K-均值算法

2.3　定位函数

本文在提取定位特征时，利用环形阵列，并结合可控功率响应方法提取多说话人的方位角，即

$$P(\theta_R)=\sum_{i=1}^{M}\sum_{j=i+1}^{M}\int_{-\infty}^{+\infty}\psi_{ij}(\omega)X_i(\omega)X_j^*(\omega)e^{j\omega\tau_{ij}(\theta_R)}d\omega \tag{5}$$

通过放置两个环形麦克风阵列，并利用这两个环形阵列来提取多说话人的定位特征（图 3）。

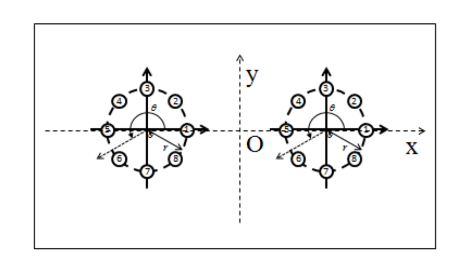


图 3　双环形阵列示意图

设坐标的原点 O 位于两个环形阵列的中间位置，两个环形阵列的中心坐标分别为 $\vec{O}_1=(x_{O_1},y_{O_1})$ 和 $\vec{O}_1=(x_{O_1},y_{O_1})$，声源的位置坐标为 $\vec{x}_s=(x,y)$，则声源对应于两个环形阵列的方位角 θ_1,θ_2 分别满足：

$$\theta_1=\arccos\left(\frac{x-x_{O_1}}{\sqrt{(x-x_{O_1})^2+(y-y_{O_1})^2}}\right) \tag{6}$$

$$\theta_2=\arccos\left(\frac{x-x_{O_2}}{\sqrt{(x-x_{O_2})^2+(y-y_{O_2})^2}}\right) \tag{7}$$

2.4　基于粒子滤波多说话人跟踪算法

本节利用并行粒子滤波算法对多说话人的状态进行跟踪，其中的运动模型采用 Langevin 模型，似然函数采用高斯似然函数，将并行粒子滤波与并行 K-均值聚类方法相结合实现多声源定位跟踪的步骤如下：

a）利用初始分布采样粒子 $\{\alpha_0^i,i=1,\cdots,N\}$，对应权值为

$\{w_0^i = 1/N, i = 1,\cdots,N\}$。

b）根据 Langevin 运动模型预测最新时刻粒子的位置分布。

c）估计每个粒子的方位角计算似然函数。方位角根据式(6)(7)计算，然后计算对应的似然函数。

d）将似然函数计算的结果作为粒子的权值，并对权值进行归一化处理，

$$\tilde{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{i=1}^{N} w_k^{(i)}} \tag{8}$$

利用归一化后的权值对粒子进行重采样。

e）针对多说话人跟踪问题，需要利用并行 K-均值聚类来确定哪些粒子来自于同一个跟踪目标，在并行 K -均值聚类时，说话人的数目已知。在聚类后说话人的位置估计为

$$\hat{\alpha}_p = \sum_{i=1}^{N_p} \alpha_{p,k}^{(i)} \tag{9}$$

其中: $\hat{\alpha}_p$ 是第的位置估计，$\alpha_{p,k}^{(i)}$ 为对应于第 p 个说话人的粒子，N_p 为粒子数目。

3 实验结果与分析

实验平台中硬件为台式机和 GTX960 显卡，软件平台为 Visual Studio 2013，实验硬件相关参数如表 2 所示。

表 2 实验硬件相关参数

GPU		CPU	
GPU	GTX960	CPU	i5-4460
核数	1024	核数	4/8
时钟率	1.3 GHz	时钟率	3.2 GHz

在粒子滤波的多声源跟踪中引入并行 K-均值聚类的目的是对粒子滤波重采样后的粒子作聚类分析，将属于同一个目标的粒子归为一类，因此实验首先给出了并行 K-均值聚类算法对粒子的聚类过程，如图 5 所示。

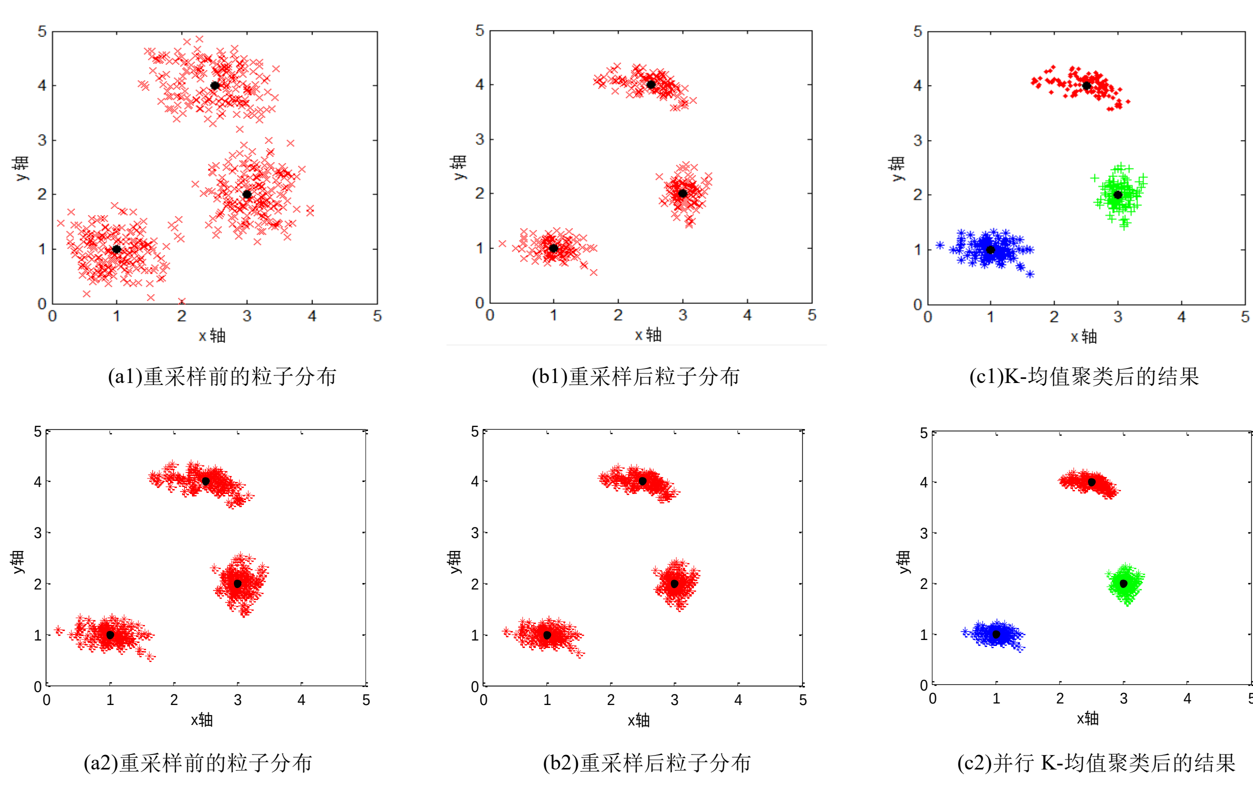


图 5 并行执行时不同阶段粒子分布的变化

从图 6 中可知，在重采样前粒子分布相对分散，而在重采样后一些权值低的粒子被大权值的粒子替换，重采样后的粒子分布相对比较集中，虽然从粒子的分布图中可以明显地观察到每个粒子所属的目标，但是粒子滤波算法本身无法对这些粒子归类，图 6(c)则给出了并行 K-均值聚类算法对粒子做分类的结果，显然属于不同目标的粒子被很明显地区分开，相比 K-均值聚类算法，并行 K-均值聚类具有更明显的区分性。

为验证本文方法在智能会议室室内场景中有多个说话人的定位跟踪效果，实验场景如图 7 所示，智能会议室的噪声及混响设置 $SNR = 20dB, T_{60} = 150ms$，放置两个环形麦克风阵列，房间中有三个说话人同时发声，运动轨迹分别为直角形、弧形和直线型，每个说话人的发声时间为 20 s。

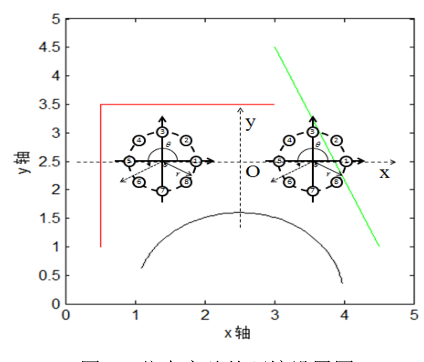
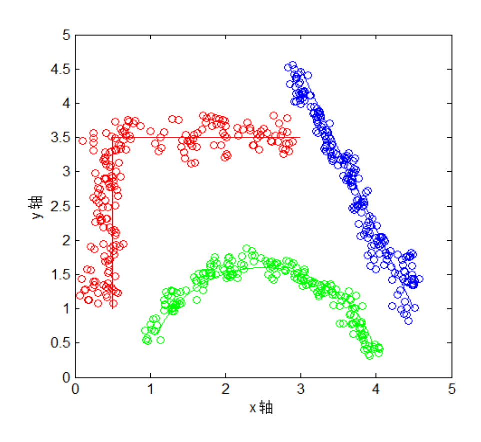
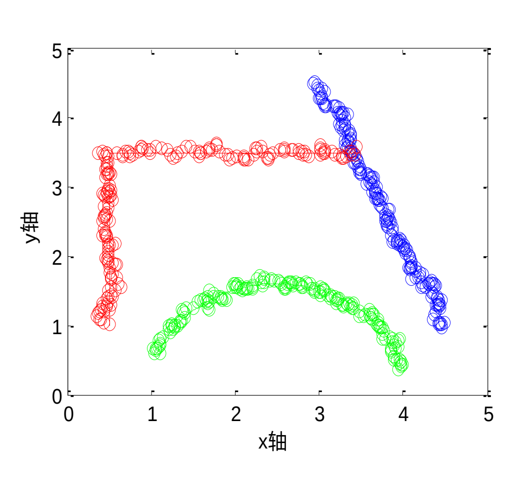


图 6 仿真实验的环境设置图



（a）粒子滤波+聚类算法的跟踪结果



（b）并行粒子滤波+聚类算法的跟踪结果

图 7　粒子滤波+聚类算法的跟踪结果

从上面实验可得出，并行粒子滤波和并行 K-均值的多说话人跟踪在声源数目的有效数目的性能比粒子滤波和 K-均值聚类的多说话人跟踪精度高。并且本文算法在高强度噪声环境下，不仅能实时确定声源有效数目，也能准确跟踪多说话人的位置。

从图 7 中可以看出，在多个说话人轨迹交叉不严重的情况下，本文方法可以跟踪到多个说话人的位置，在说话人数目已知的情况下，并行 K-均值聚类算法可以更有效地对重采样粒子进行划分，从而得到了精度较高的多说话人位置估计结果。图 8 和 9 分别给出了 x 轴和 y 轴的跟踪轨迹。

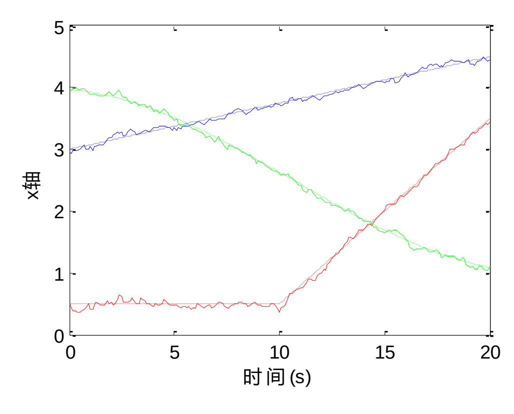


图 8　X 轴的跟踪结果

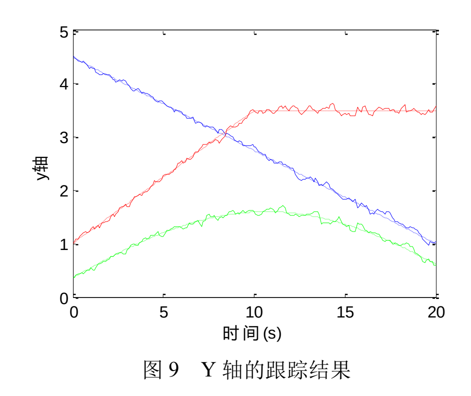


图 9　Y 轴的跟踪结果

表 3　两种算法在 X 方向的跟踪误差比较

算法	蓝	绿	红
PF+K 均值跟踪误差率 (%)	0.9015	1.5046	13.112
本文算法跟踪误差率 (%)	0.8052	0.7493	5.475

从图 8、9 中分别选出前 10 个时刻的 X 轴的位置，作误差率比较，即用跟踪得到的位置与说话人的真实位值取相对误差值，分别用图 8 中的蓝绿红三条轨迹作误差比较，得出结果如表 3 所示。根据表 3 中误差率可知，本文算法的误差率都比传统粒子滤波和 K-均值聚类的跟踪算法误差率低，即本文算法的跟踪精度高于传统跟踪算法，本文算法性能更好。

综上所述，并行粒子滤波和基于 GPU 的 K-均值聚类算法的多说话人跟踪在声源数目的有效数目的性能比粒子滤波和 K-均值聚类的多说话人跟踪效果好。并且本文算法在高强度噪声环境下，不仅能实时确定声源有效数目，也能准确跟踪多说话人的位置。

4　结束语

通过分析粒子滤波在多说话人跟踪中存在的问题，本文通过引入并行策略，将粒子滤波算法并行实现，降低算法复杂度。并分析了多个目标出现在跟踪区域时，粒子出现的发散的问题，采用基于 GPU 的 K-均值聚类的方法来解决。实验结果表明，本文算法有效提高算法在 GPU 中硬件资源利用率，粒子收敛性更强，本文算法最终的多目标跟踪中跟踪精度也比原始粒子滤波跟踪算法高。

参考文献：

[1] 曹洁，余丽珍. 基于 MFCC 和运动强度聚类初始化的多说话人识别 [J]. 计算机应用研究, 2012, 29 (9): 3295-3298.

[2] 屈丹，张文林. 基于本征音子说话人子空间的说话人自适应算法 [J]. 电子与信息学报, 2015, 37 (6): 1350-1356.

[3] 张微，康宝生. 相关滤波目标跟踪进展综述 [J]. 中国图象图形学报, 2017, 22 (8): 1017-1033.

[4] 任航. 基于拟蒙特卡洛滤波的改进式粒子滤波目标跟踪算法 [J]. 电子测量与仪器学报, 2015 (2): 289-295.

[5] 秦永元，张洪钺，汪叔华. 卡尔曼滤波与组合导航原理 [M]. 3 版. 西安:

西北工业大学出版社, 2015.
[6] 翟卫欣, 程承旗. 基于 Kalman 滤波的 Camshift 运动跟踪算法 [J]. 北京大学学报: 自然科学版, 2015, 51 (5): 799-804.
[7] 程兰, 王志远, 陈杰, 等. 基于粒子滤波和滑动平均扩展 Kalman 滤波的多径估计算法 [J]. 电子与信息学报, 2017, 39 (3): 709-716.
[8] 雷明, 韩崇昭, 肖梅. 扩展卡尔曼粒子滤波算法的一种修正方法 [J]. 西安交通大学学报, 2005, 39 (8): 824-827.
[9] 张应博. 基于无极卡尔曼滤波算法的雅可比矩阵估计 [J]. 计算机应用, 2011, 31 (6): 1699-1702.
[10] 李天成, 范红旗, 孙树栋. 粒子滤波理论、方法及其在多目标跟踪中的应用 [J]. 自动化学报, 2015, 41 (12): 1981-2002.
[11] 林静, 杨继臣, 张雪源, 等. 基于稀疏表示权重张量的音频特征提取算法 [J]. 计算机应用, 2016, 36 (5): 1426-1429.
[12] 孙海洋, 张利. 无人机跟踪场景下的粒子滤波算法的改进 [J]. 计算机仿真, 2017, 34 (2): 84-87.
[13] Goyal B, Budhraja T, Bhatnagar R, et al. Implementation of Particle Filters for Single Target Tracking Using CUDA [C]// Proc of the 5th International Conference on Advances in Computing and Communications. 2016: 28-32.
[14] Baydoun M, Dawi M, Ghaziri H. Enhanced parallel implementation ofthe K-Means clustering algorithm [C]// Proc of International Conference on Advances in Computational TOOLS for Engineering Applications. 2016: 7-11.
[15] 李晓瑜, 俞丽颖, 雷航, 等. 一种 K-means 改进算法的并行化实现与应用 [J]. 电子科技大学学报, 2017, 46 (1): 61-68.
[16] 陈平华, 陈传瑜. 基于满二叉树的二分 K-means 聚类并行推荐算法 [J]. 计算机工程与科学, 2015, 37 (8): 1450-1457.